Reinforcement Learning Assignment 1 Report

Environment Details:

1. The environment used in this assignment consists of a single agent.
2. Deterministic environment: Given an action step, the agent moves to the next state with 100% probability.
3. Stochastic environment: Given an action step, the agent moves to the next state with 90% probability. 10% of the time, it stays in the same state.
4. **States:** The environment used in this assignment is a 4 by 4 grid and there are 16 states in the environment from (S1 to S16).
5. The agent's initial position is at (0,0) and the Goal state is at (3,3).
6. **Actions:** Four actions (up,down,right,left) are defined at each state. (0-up,1-down,2-right,3-left).
7. **Rewards:** 5 positional rewards are present in the environment, out of which 2 are positive rewards,2 are negative rewards and 1 is the positive reward for reaching the goal state.
8. The positive rewards present at (1,1) *(Collect Coffee)* and (2,2) (*Do assignment)* have a reward value of +3 and +6 respectively. Similarly the negative rewards present at (2,0) and (0,2) *(Bunking a Class)* have a reward value of -5 and -6 respectively. The reward for reaching the goal state is +10 *(Reach RL class)*.
9. **Objective:** The goal of the agent is to reach the Goal state in a minimum number of steps with maximum reward. The agent should collect positive rewards while discarding negative rewards.

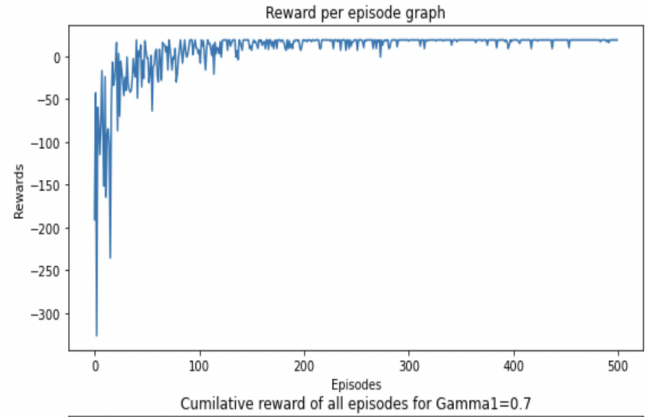
Environment Visualization:

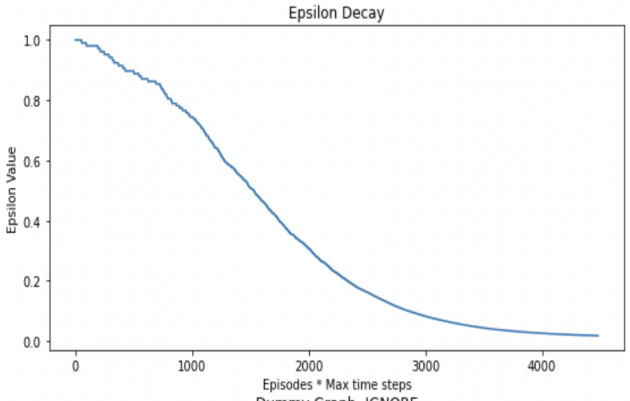
Q-Learning for solving the Deterministic Environment:

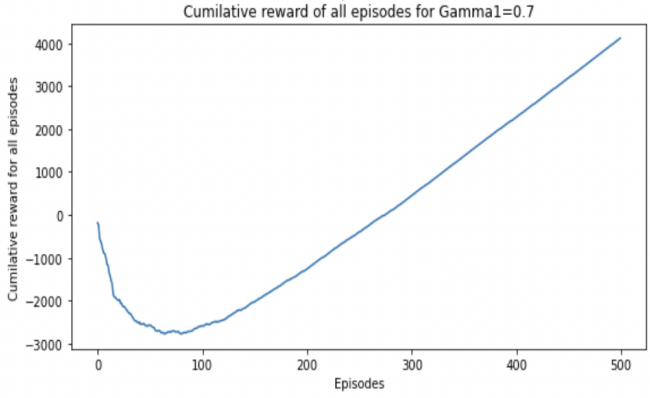
1. Q-Learning applied to the environment described in the previous heading.
2. Agent successfully moves from the Initial state to the Goal state by collecting the maximum rewards.
3. We use exponential decay here, to decay our epsilon value.
4. The parameters used to train the agent are:
    a. Learning Rate- alpha=0.4
    b. Discount Rate-Gamma=0.7
    c. Number of episodes=500
    d. Maximum number of time steps=100
    e. Epsilon=1
    f. Maximum Epsilon=1
    g. Minimum Epsilon=0.01
    h. Epsilon Decay Rate= 0.01
5. The agent is trained for 500 episodes each time calculating the Qtable values and updating them using the function :

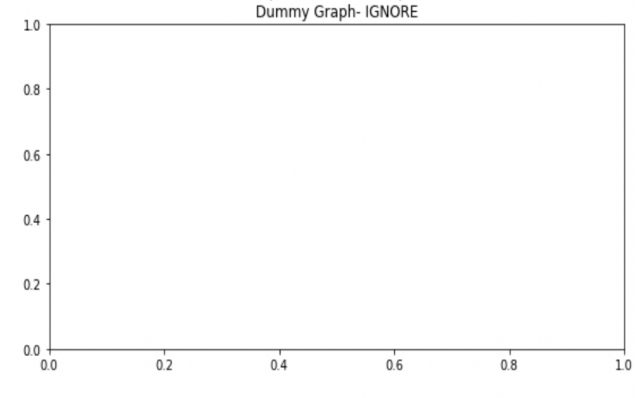
$$Q^{new}(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\Big(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} - \underbrace{Q(s_t, a_t)}_{\text{old value}}\Big)}^{\text{temporal difference}}$$
$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{new value (temporal difference target)}}$$

Training Graphs for Q-Learning Deterministic Environment:









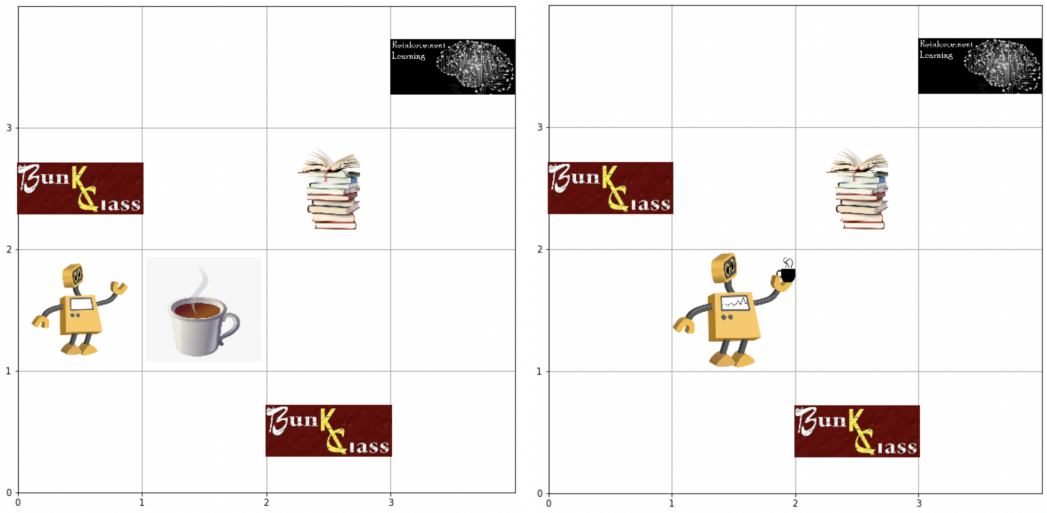
The graphs presented above are for Q-Learning Deterministic environments.

1. **Plot1:**
    a. Defines the cumulative reward per episode graph. The X-axis denotes the number of episodes and Y axis denotes the cumulative reward per episode.
    b. From the graph we can infer that the model converges around 75 episodes. The cumulative rewards before around 75 episodes are fluctuating a lot, once the agent learns the policy, the rewards per episode are stable.
2. **Plot 2:**
    a. The Line graph in plot 2 indicates the epision decay with respect to exponential decay.
    b. The Y axis are the Epsilon values that start at 1 (also the max epsilon value) and gradually tapers off at 0.01 (min epsilon value)
3. **Plot 3:**
    a. Graph in plot3 shows that, the cumulative of the cumulative reward per episode decreases first as the agent is exploring the environment and is collecting some negative rewards.
    b. But as the number of episodes passes, the cumulative of the cumulative reward per episode increases linearly.

Testing Visualization for Q-Learning Deterministic Environment: (Read from left to right)

Upon obtaining the Qtable after running the 500 episodes mentioned above, This Qtable can be passed to the agent to learn the optimal policy and move to the goal state with maximum reward.

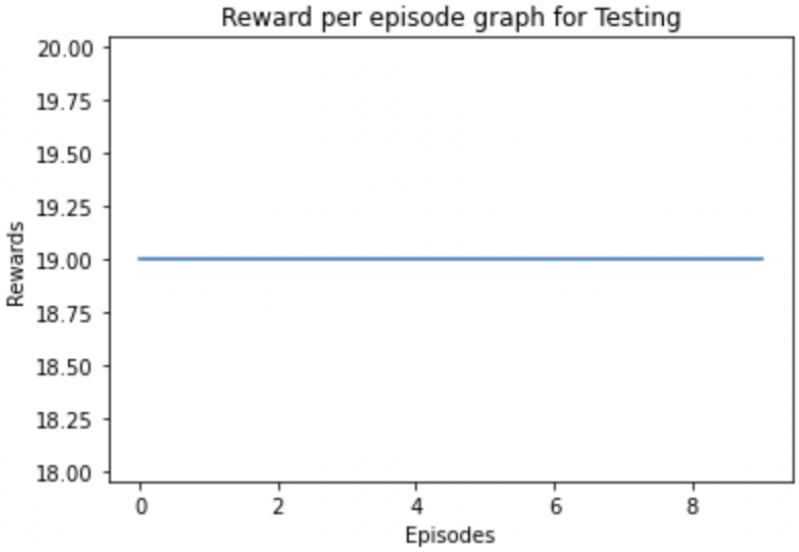
Results:



In the above images, we can see that the agent moves from (0,0) to (1,0) as shown in figure 1. Then proceeds to move to the goal states in the least number of steps while avoiding the negative rewards and collecting the positive rewards.

Testing Graphs for Q-Learning Deterministic Environment:

Using the optimal policy learnt, the agent was tested for 10 episodes. At each episode the cumulative reward was 19 (maximum possible reward in a deterministic environment).

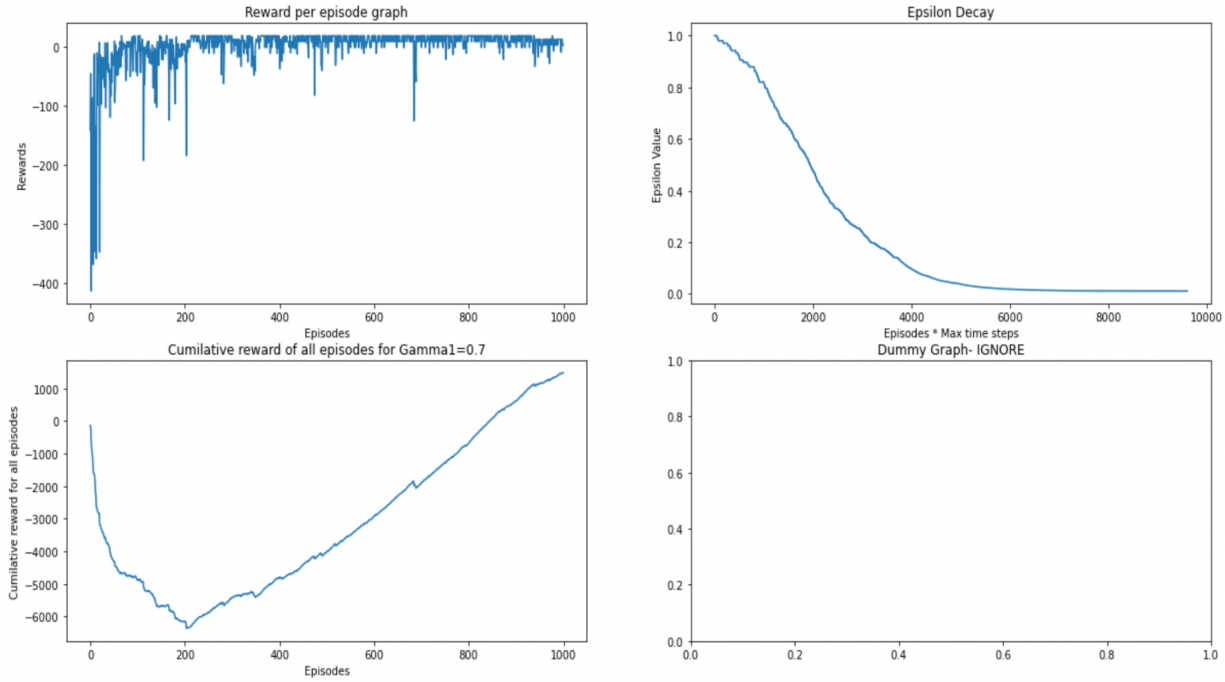


Q-Learning for solving the Stochastic Environment:

6. Q-Learning applied to the environment previously described.
7. Agent successfully moves from the Initial state to the Goal state by collecting the maximum rewards.
8. We use exponential decay here, to decay our epsilon value.
9. The parameters used to train the agent are:
    i. Learning Rate- alpha=0.4
    j. Discount Rate-Gamma=0.7
    k. Number of episodes=1000
    l. Maximum number of time steps=100
    m. Epsilon=1
    n. Maximum Epsilon=1
    o. Minimum Epsilon=0.01
    p. Epsilon Decay Rate= 0.01
10. The agent is trained for 500 episodes each time calculating the Qtable values and updating them using the function :

$$Q^{new}(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\Big(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}^{\text{new value (temporal difference target)}} - \underbrace{Q(s_t, a_t)}_{\text{old value}}\Big)}^{\text{temporal difference}}$$

Training Graphs for Q-Learning Stochastic Environment:



The graphs presented above are for Q-Learning Stochastic environments.

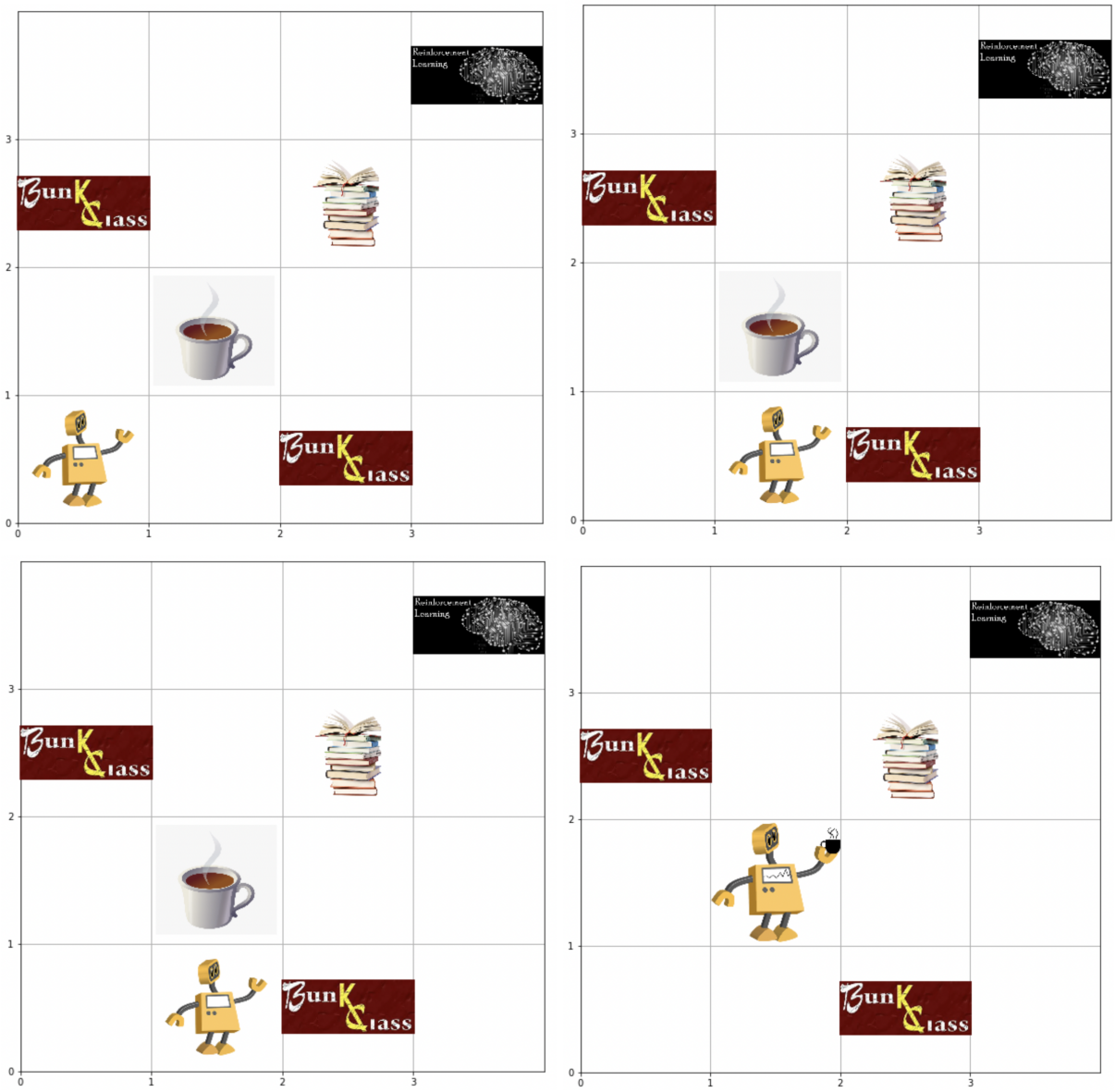
4. **Plot1:**
    a. Defines the cumulative reward per episode graph. The X-axis denotes the number of episodes and Y axis denotes the cumulative reward per episode.
    b. From the graph we can infer that the model converges around 200 episodes for Stochastic Environment. The cumulative rewards before around 200 episodes are fluctuating a lot, once the agent learns the policy, the rewards per episode are stable.
    c. Some inconsistencies in cumulative reward per episode can be found, as the environment is stochastic and staying in the same state due to stochasticity is not ideal as it increases the number of time steps taken to reach the goal. Hence this negatively impacts the graph.
5. **Plot 2:**
    a. The Line graph in plot 2 indicates the epision decay with respect to exponential decay.
    b. The Y axis are the Epsilon values that start at 1 (also the max epsilon value) and gradually tapers off at 0.01 (min epsilon value)
6. **Plot 3:**
    a. Graph in plot3 shows that, the cumulative of the cumulative reward per episode decreases first as the agent is exploring the environment and is collecting some negative rewards.

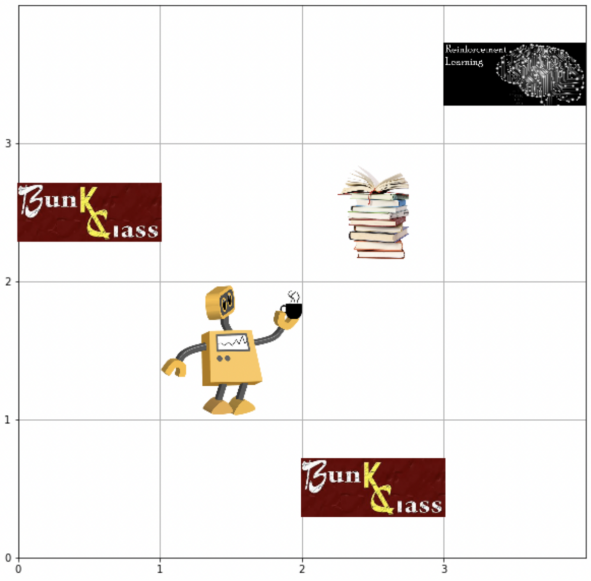
b. But as the number of episodes passes, from around episode 200 the cumulative of the cumulative reward per episode increases linearly.

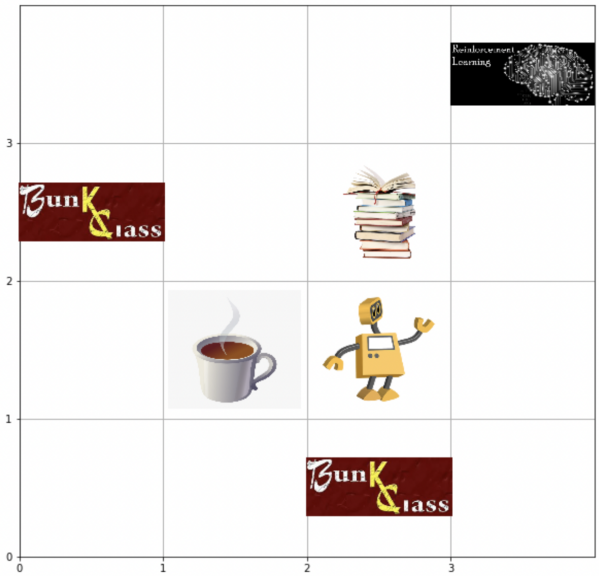
Testing Visualization for Q-Learning Stochastic Environment: (Read from left to right)

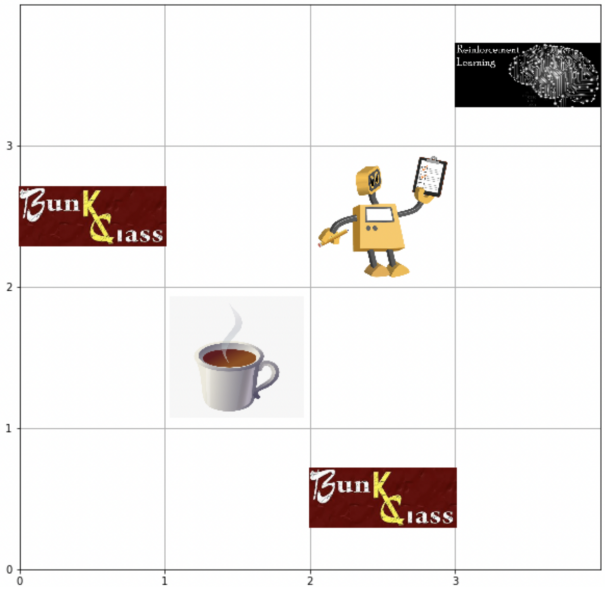
Upon obtaining the Qtable after running the 2000 episodes mentioned above, This Qtable can be passed to the agent to learn the optimal policy and move to the goal state with maximum reward.

Results:

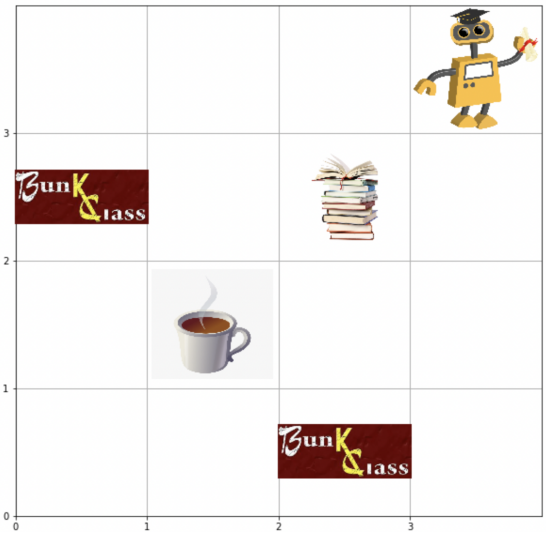



Reinforcement Learning
BunK Class
BunK Class
0
1
2
3


Reinforcement Learning
BunK Class
BunK Class
0
1
2
3


Reinforcement Learning
BunK Class
BunK Class
0
1
2
3

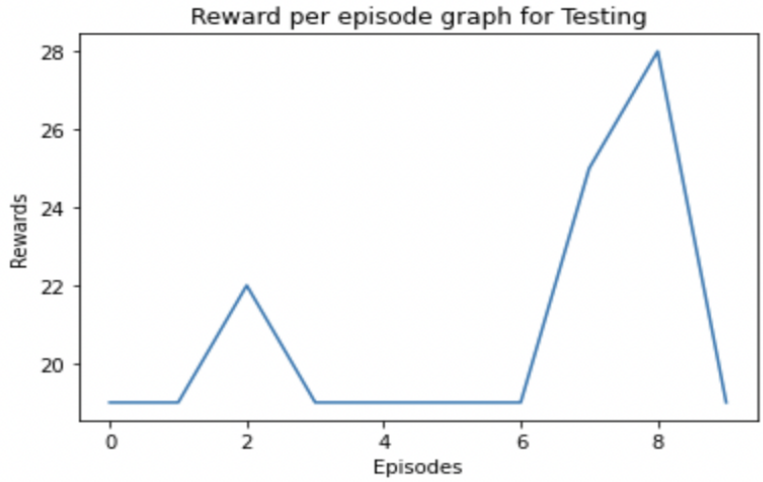

Reinforcement Learning
BunK Class
BunK Class
0
1
2
3



In the above images, we can see that the agent moves from (0,0) to (1,0) as shown in figure 1. Then proceeds to move to the goal states in the least number of steps while avoiding the negative rewards and collecting the positive rewards.

But different from the deterministic environment, the agent is stuck in state (0,2) for 2 episodes. The agent is further stuck at (1,1) when collecting positive reward 1. Hence the cumulative reward for that episode changes, as positive reward 1 is being collected twice. The agent takes 8 steps here compared to 6 steps in the deterministic environment.
Hence this stochasticity induced can result in a varied number of time steps taken to reach the goal and the cumulative rewards collected per episode.

Testing Graphs for Q-Learning Stochastic Environment:



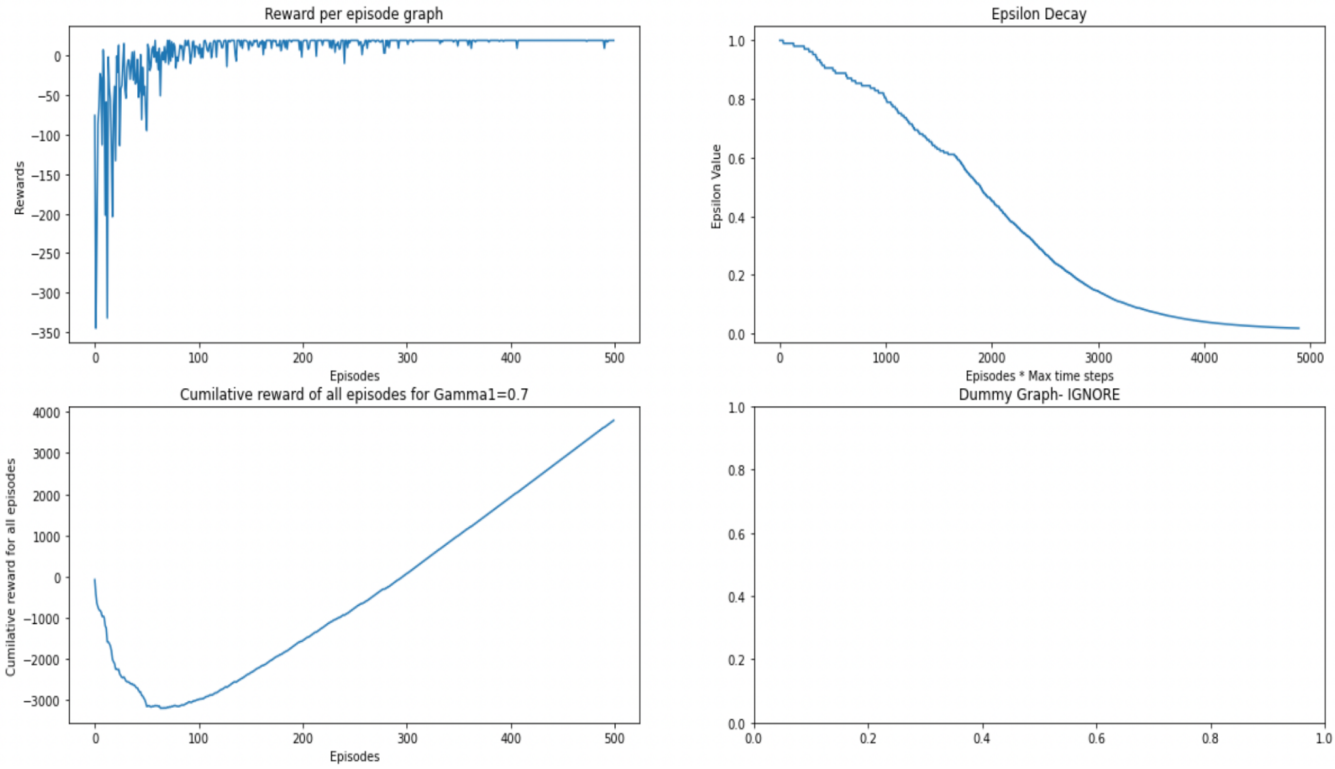
Using the optimal policy learnt, the agent was tested for 10 episodes. At each episode the cumulative reward varied for the stochastic environment. We can observe at episode 2 reward was 22, at, episode 7 reward is 26 and finally at step 8 reward was 28. Remaining all episodes had an expected reward of 19.

SARSA for solving the Deterministic Environment:

1. SARSA applied to the environment described in the previous heading.
2. Agent successfully moves from the Initial state to the Goal state by collecting the maximum rewards.
3. We use exponential decay here, to decay our epsilon value.
4. The parameters used to train the agent are:
    a. Learning Rate- alpha=0.4
    b. Discount Rate-Gamma=0.7
    c. Number of episodes=500
    d. Maximum number of time steps=100
    e. Epsilon=1
    f. Maximum Epsilon=1
    g. Minimum Epsilon=0.01
    h. Epsilon Decay Rate= 0.01
5. The agent is trained for 500 episodes each time calculating the Qtable values and updating them using the function :

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right]$$

Training Graphs for SARSA Deterministic Environment:



The graphs presented above are for SARSA Deterministic environments.

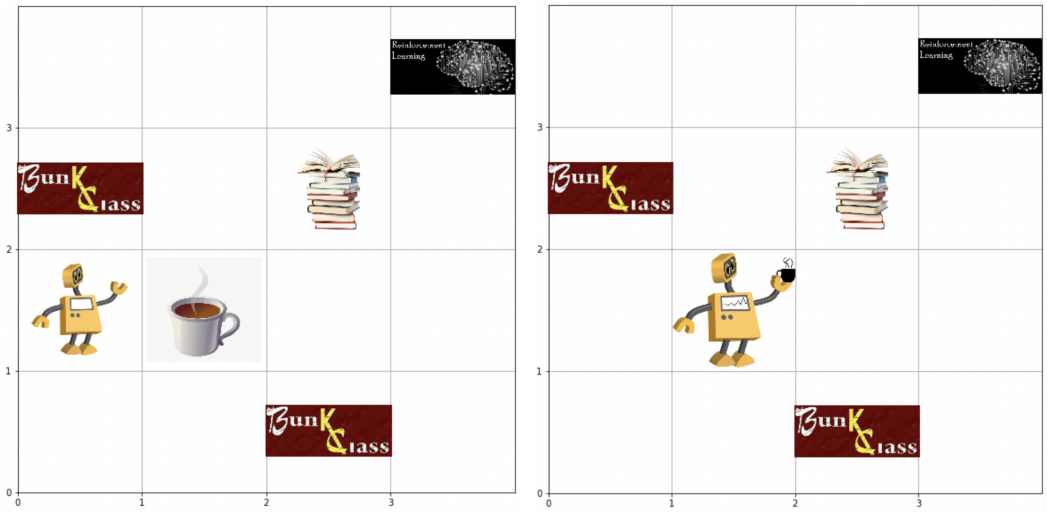
7. **Plot1:**
    c. Defines the cumulative reward per episode graph. The X-axis denotes the number of episodes and Y axis denotes the cumulative reward per episode.
    d. From the graph we can infer that the model converges around 80 episodes. The cumulative rewards before around 80 episodes are fluctuating a lot, once the agent learns the policy, the rewards per episode are stable.
8. **Plot 2:**
    c. The Line graph in plot 2 indicates the epision decay with respect to exponential decay.
    d. The Y axis are the Epsilon values that start at 1 (also the max epsilon value) and gradually tapers off at 0.01 (min epsilon value)
9. **Plot 3:**
    c. Graph in plot3 shows that, the cumulative of the cumulative reward per episode decreases first as the agent is exploring the environment and is collecting some negative rewards.

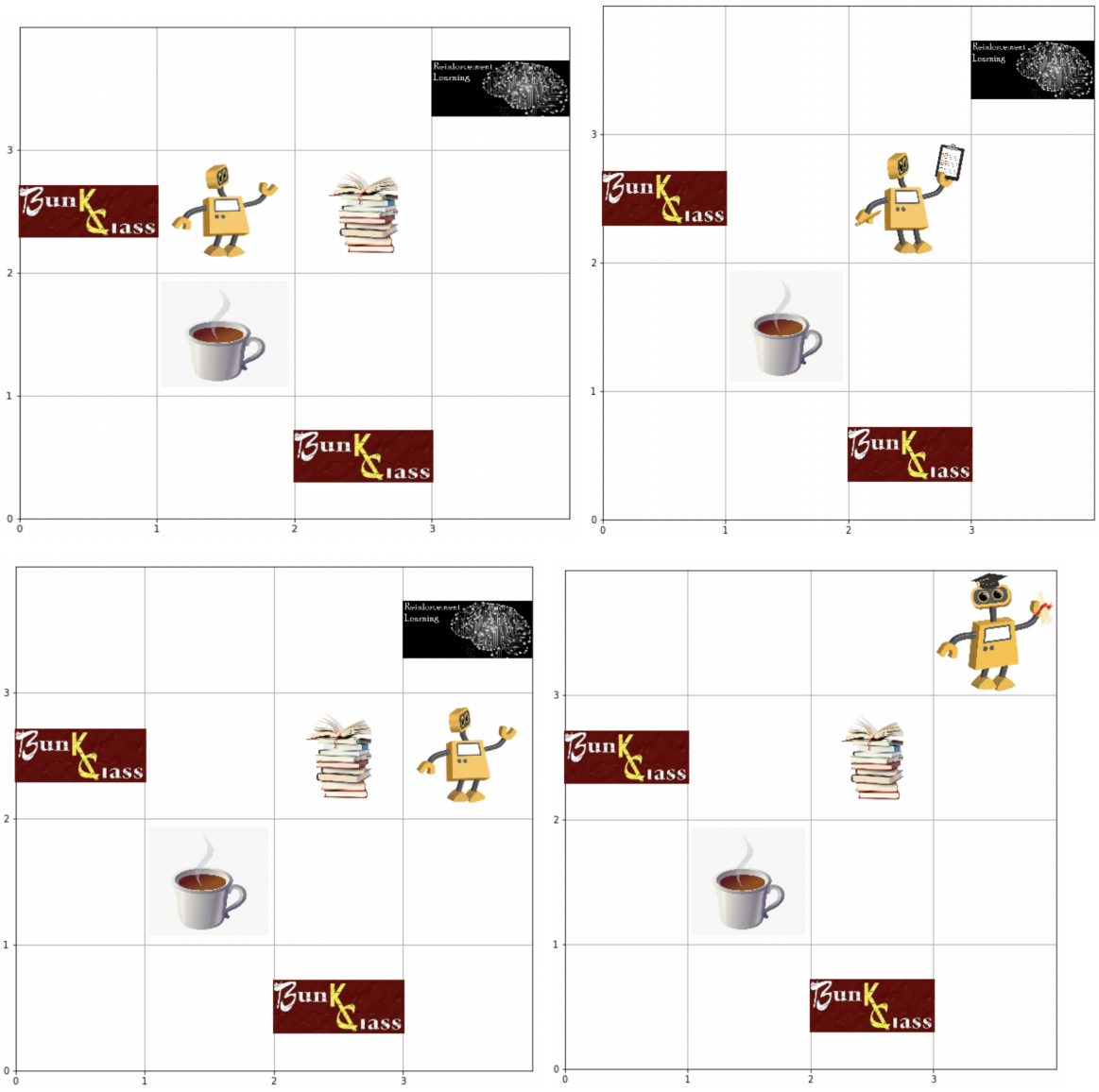
d. But as the number of episodes passes, the cumulative of the cumulative reward per episode increases linearly.

Testing Visualization for SARSA Deterministic Environment: (Read from left to right)

Upon obtaining the Qtable after running the 500 episodes mentioned above, This Qtable can be passed to the agent to learn the optimal policy and move to the goal state with maximum reward.

Results:

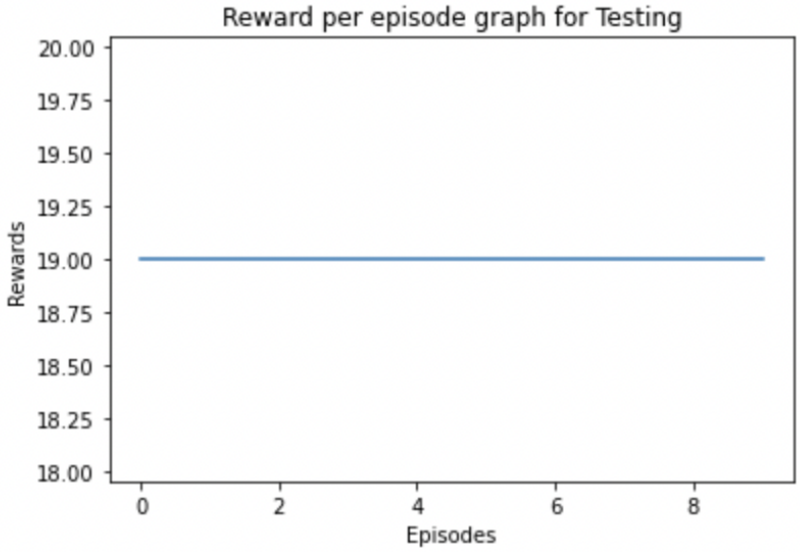




In the above images, we can see that the agent moves from (0,0) to (1,0) as shown in figure 1. Then proceeds to move to the goal states in the least number of steps while avoiding the negative rewards and collecting the positive rewards.

Testing Graphs for SARSA Deterministic Environment:

Using the optimal policy learnt, the agent was tested for 10 episodes. At each episode the cumulative reward was 19 (maximum possible reward in a deterministic environment).

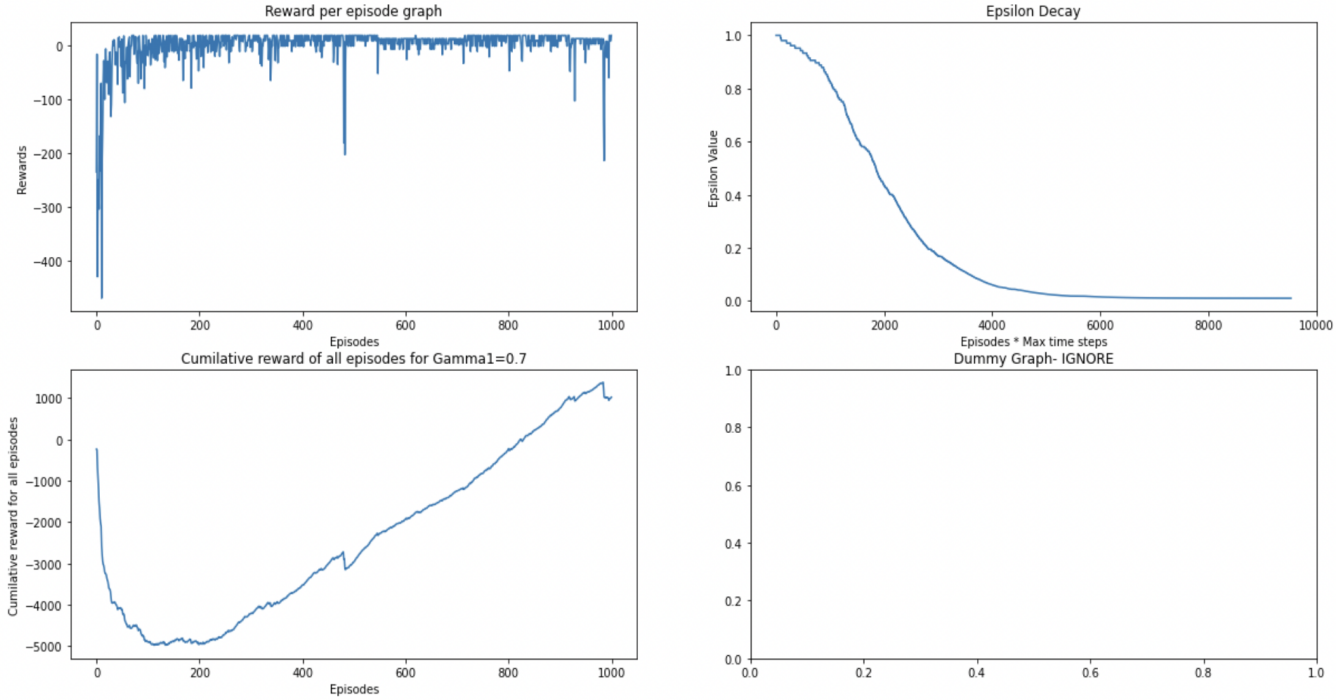


SARSA for solving the Stochastic Environment:

11. SARSA applied to the environment previously described.
12. Agent successfully moves from the Initial state to the Goal state by collecting the maximum rewards.
13. We use exponential decay here, to decay our epsilon value.
14. The parameters used to train the agent are:
    q. Learning Rate- alpha=0.4
    r. Discount Rate-Gamma=0.7
    s. Number of episodes=1000
    t. Maximum number of time steps=100
    u. Epsilon=1
    v. Maximum Epsilon=1
    w. Minimum Epsilon=0.01
    x. Epsilon Decay Rate= 0.01
15. The agent is trained for 500 episodes each time calculating the Qtable values and updating them using the function :

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right]$$

Training Graphs for SARSA Stochastic Environment:



The graphs presented above are for Q-Learning Stochastic environments.

10. **Plot1:**
    d. Defines the cumulative reward per episode graph. The X-axis denotes the number of episodes and Y axis denotes the cumulative reward per episode.
    e. From the graph we can infer that the model converges around 150 episodes for Stochastic Environment. The cumulative rewards before around 150 episodes are fluctuating a lot, once the agent learns the policy, the rewards per episode are stable.
    f. Some inconsistencies in cumulative reward per episode can be found, as the environment is stochastic and staying in the same state due to stochasticity is not ideal as it increases the number of time steps taken to reach the goal. Hence this negatively impacts the graph.
11. **Plot 2:**
    c. The Line graph in plot 2 indicates the epision decay with respect to exponential decay.
    d. The Y axis are the Epsilon values that start at 1 (also the max epsilon value) and gradually tapers off at 0.01 (min epsilon value)
12. **Plot 3:**
    c. Graph in plot3 shows that, the cumulative of the cumulative reward per episode decreases first as the agent is exploring the environment and is collecting some negative rewards.

d. But as the number of episodes passes, from around episode 150 the cumulative of the cumulative reward per episode increases linearly.

Testing Visualization for SARSA Stochastic Environment: (Read from left to right)

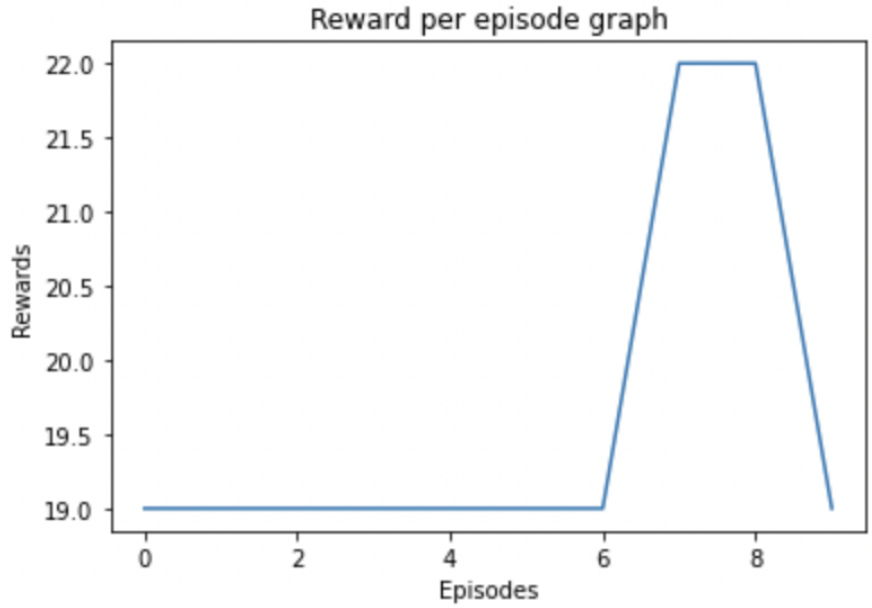
Upon obtaining the Qtable after running the 2000 episodes mentioned above, This Qtable can be passed to the agent to learn the optimal policy and move to the goal state with maximum reward.

Results:

In the above images, we can see that the agent moves from (0,0) to (1,0) as shown in figure 1. Then proceeds to move to the goal states in the least number of steps while avoiding the negative rewards and collecting the positive rewards.

But different from the deterministic environment, the agent is stuck in state (0,0) for 2 episodes. The agent is further stuck at (2,1) when collecting positive reward 1. Hence the cumulative reward for that episode changes, as positive reward 1 is being collected twice. The agent takes 8 steps here compared to 6 steps in the deterministic environment.
Hence this stochasticity induced can result in a varied number of time steps taken to reach the goal and the cumulative rewards collected per episode.

Testing Graphs for SARSA Stochastic Environment:



Using the optimal policy learnt, the agent was tested for 10 episodes. At each episode the cumulative reward varied for the stochastic environment. We can observe at episode 7 and 8 the reward was 22. Remaining all episodes had an expected reward of 19.

Comparing the reward per episode graphs of Q-Learning and SARSA:

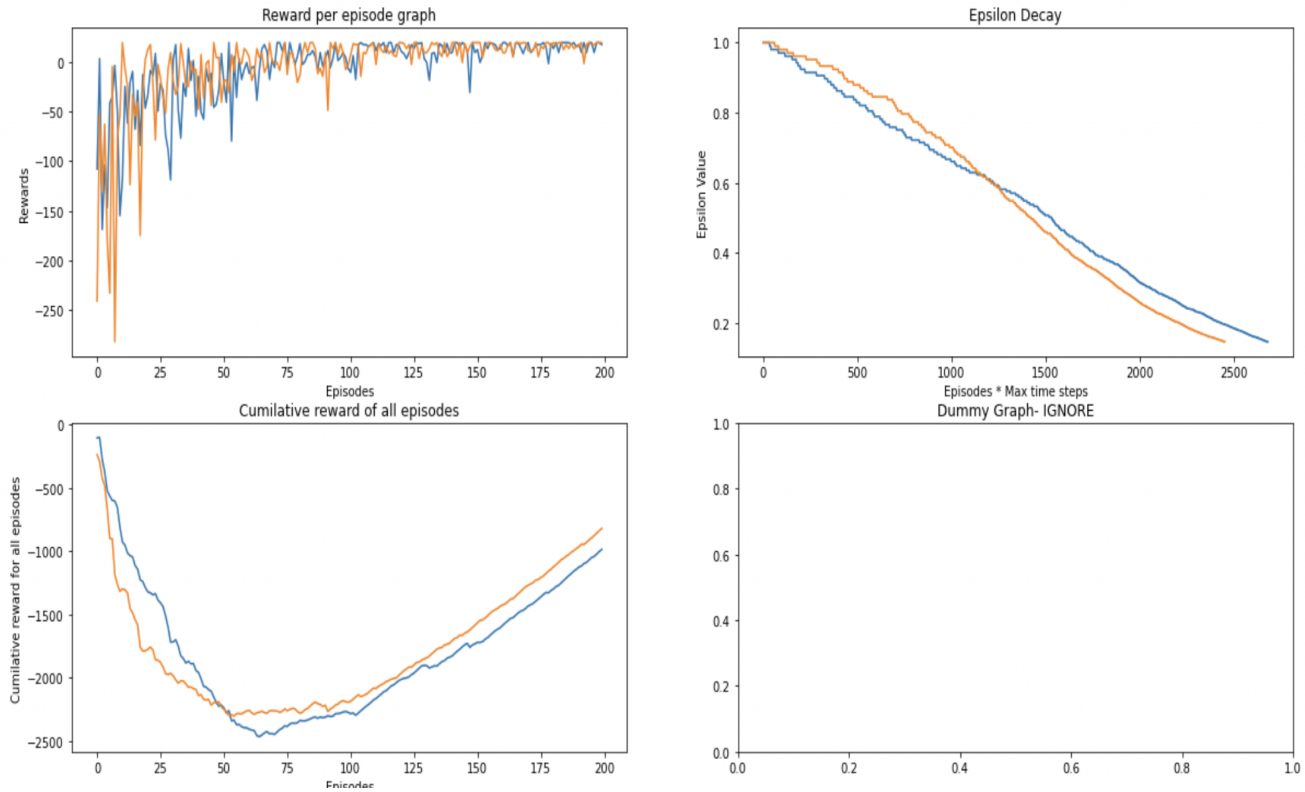
A. Deterministic Environment:

Both Q-learning and SARSA were run on our 4 by 4 grid environment, first case is for the Deterministic Environment.

1. The parameters used by both Q-learning and SARSA to train the agent are:
    i. Learning Rate- alpha=0.4
    j. Discount Rate-Gamma=0.7
    k. Number of episodes=200
    l. Maximum number of time steps=100
    m. Epsilon=1
    n. Maximum Epsilon=1
    o. Minimum Epsilon=0.01
    p. Epsilon Decay Rate= 0.01

2. The agent is trained for 200 episodes each time calculating the Qtable values for both Q-Learning and SARSA separately.

Comparison Results:



1. Here the blue line on the graph indicates results for SARSA.
2. The orange line on the graph indicates results for Q-Learning.
3. From the cumulative rewards per episode Graph (graph-3), we can see that Q-learning (Orange line) converges at around 50 episodes, while SARSA (blue-line) converges at around 60 episodes.
4. Hence, we can infer that for this environment, when the environment is deterministic, Q-Learning has a slight edge over SARSA.

B. Stochastic Environment:

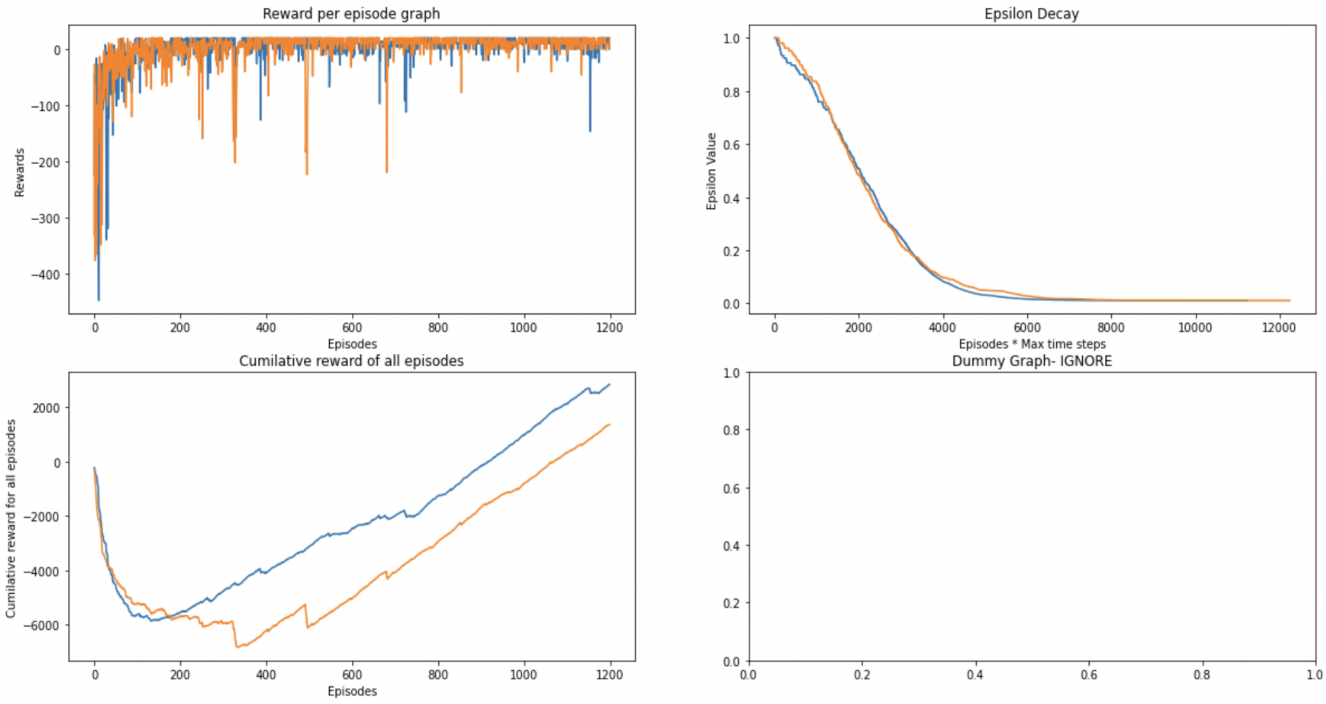
Both Q-learning and SARSA were run on our 4 by 4 grid environment, first case is for the Deterministic Environment.

1. The parameters used by both Q-learning and SARSA to train the agent are:
    a. Learning Rate- alpha=0.4
    b. Discount Rate-Gamma=0.7
    c. Number of episodes=1200
    d. Maximum number of time steps=100
    e. Epsilon=1
    f. Maximum Epsilon=1

g. Minimum Epsilon=0.01
h. Epsilon Decay Rate= 0.01

2. The agent is trained for 1200 episodes each time calculating the Qtable values for both Q-Learning and SARSA separately.

Comparison Results:



1. Here the blue line on the graph indicates results for SARSA.
2. The orange line on the graph indicates results for Q-Learning.
3. From the cumulative rewards per episode Graph (graph-3), we can see that Q-learning (Orange line) converges at around 300 episodes, while SARSA (blue-line) converges at around 150 episodes.
4. Hence, we can infer that for this environment, when the environment is stochastic, SARSA is better than Q-Learning.

Tabular Methods Explanation:

Q-Learning:

1. Q-Learning is a reinforcement learning algorithm that is used to determine the value of an action for a given state.
2. It is a Model free reinforcement learning algorithm. Ie. it doesnt need the model of the environment.
3. Q learning gets the optimal policy taking the maximum expected value of the total rewards over other steps from the current state.
4. Q-learning formula used in this assignment is:

qtable_state[action]=qtable_state[action]+(alpha(reward+gamma*max(env.qtable[new_observation])-qtable_state[[action]]))*

SARSA:

1. SARSA stands for State action reward state action. Which is a reinforcement learning algorithm.
2. SARSA is also a model free reinforcement learning algorithm like Q-learning.
3. In SARSA after we calculate the next step and obtain the reward, we use a next action A' to calculate the Q table values.
4. Qvalue depends on the current state of the agent S and the action A, On performing step function, we get a new S'. We generate a new action A'.
5. The SARSA algorithm formula used in this assignment is:

qtable_state[action]=qtable_state[action]+(alpha(reward+gamma*(env.qtable[new_observation][new_action])-qtable_state[[action]]))*

SAFETY IN AI:

Safety in AI is an important aspect to be taken into consideration. A safe AI agent doesn't encounter system failures, damage itself or cause harm to the environment. Robots must be able to safely learn and adapt online to solve their assigned tasks effectively and with high performance. We should ensure that this automatic adaption happens safely. Ie. If the agent takes an exploratory action that current policy should be able to recover.

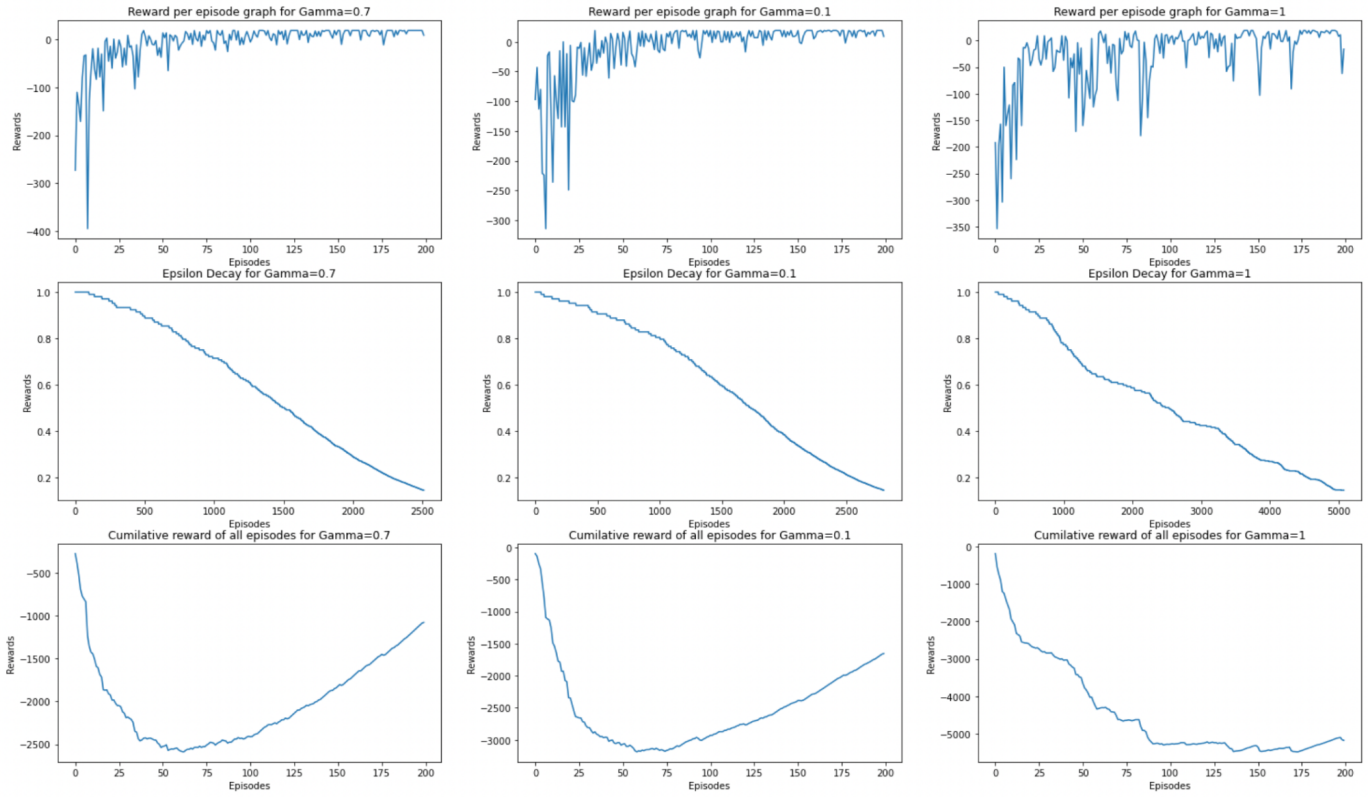
Steps to ensure safety of environment:

- Explicitly model and learn about uncertainties in the robots dynamics and its environments.
- Develop algorithms for safe exploring and safe policy updates.
- Secure the code and prevent it from being hacked and tampered.
- Encryption of data if an agent is communicating with other agents or systems.
- Comprehend all possible threats and enable good design and implementation changes to secure the application.

BONUS : HYPERPARAMETER Testing

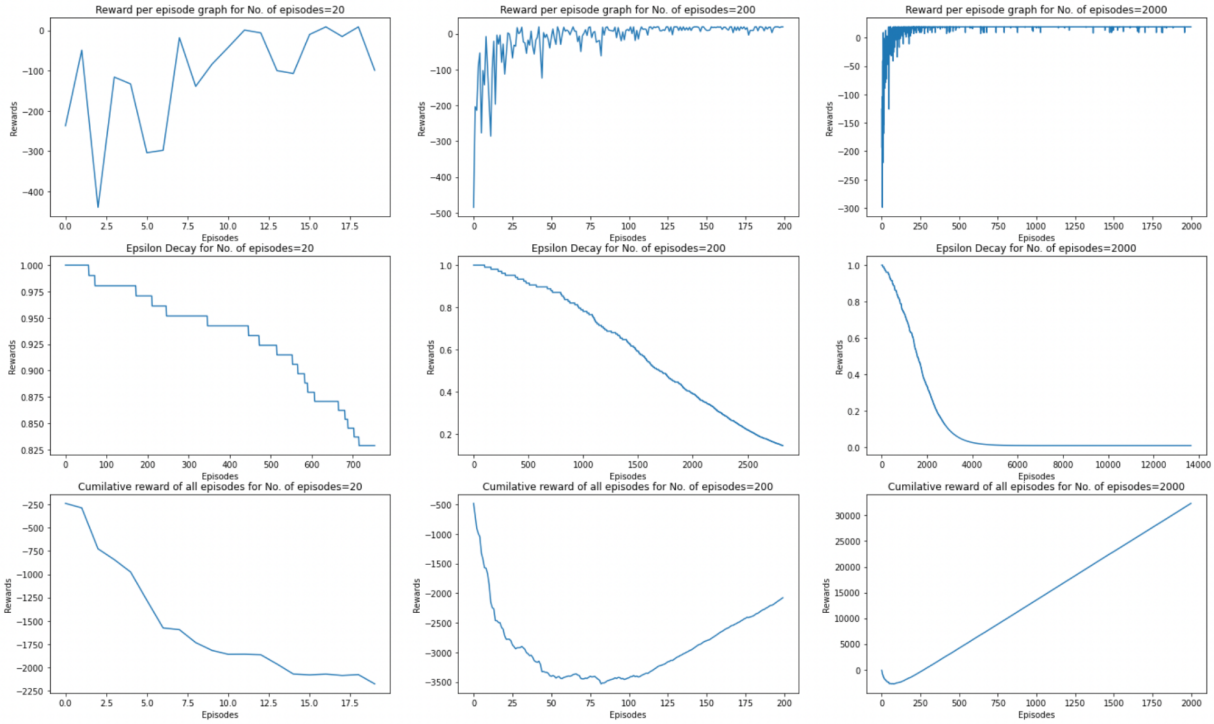
A. Q-Learning Deterministic Environment (Varied GAMMA)

Changing Gamma values (gamma1=0.7, gamma2=0.1, gamma3=1)



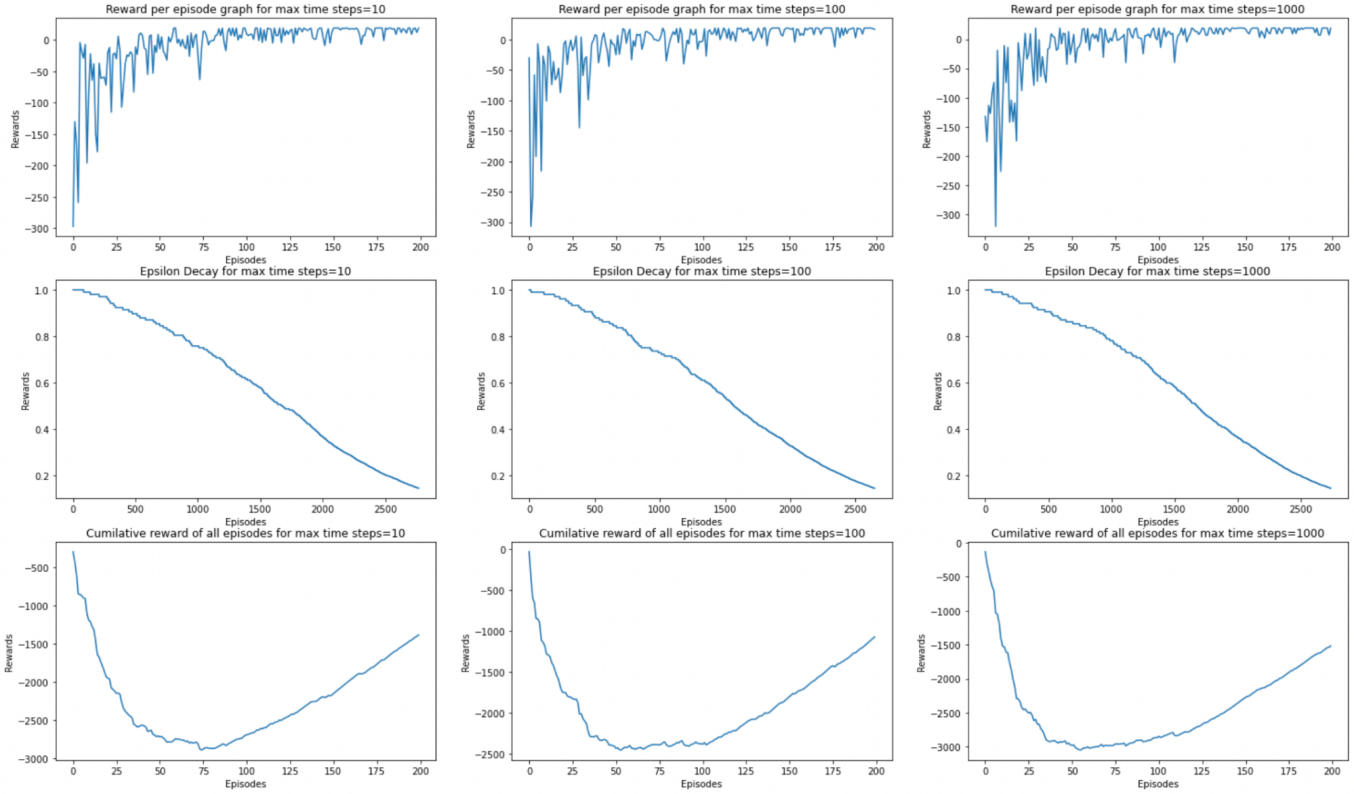
1. For the given environment we have executed the Q-learning algorithm in a deterministic environment.
2. The Column 1 of the plots show the reward per episode, Epsilon decay and Cumulative reward for gamma=0.7. Similarly column 2 and column 3 are for gamma values 0.1 and 1 respectively.
3. From the graphs we can notice that the best value for Gamma is 0.7. Take a look at row 1 rewards per episode graph, it produces the best results for gamma=0.7

B. Q-Learning Deterministic Environment (Varied Number of Episodes)



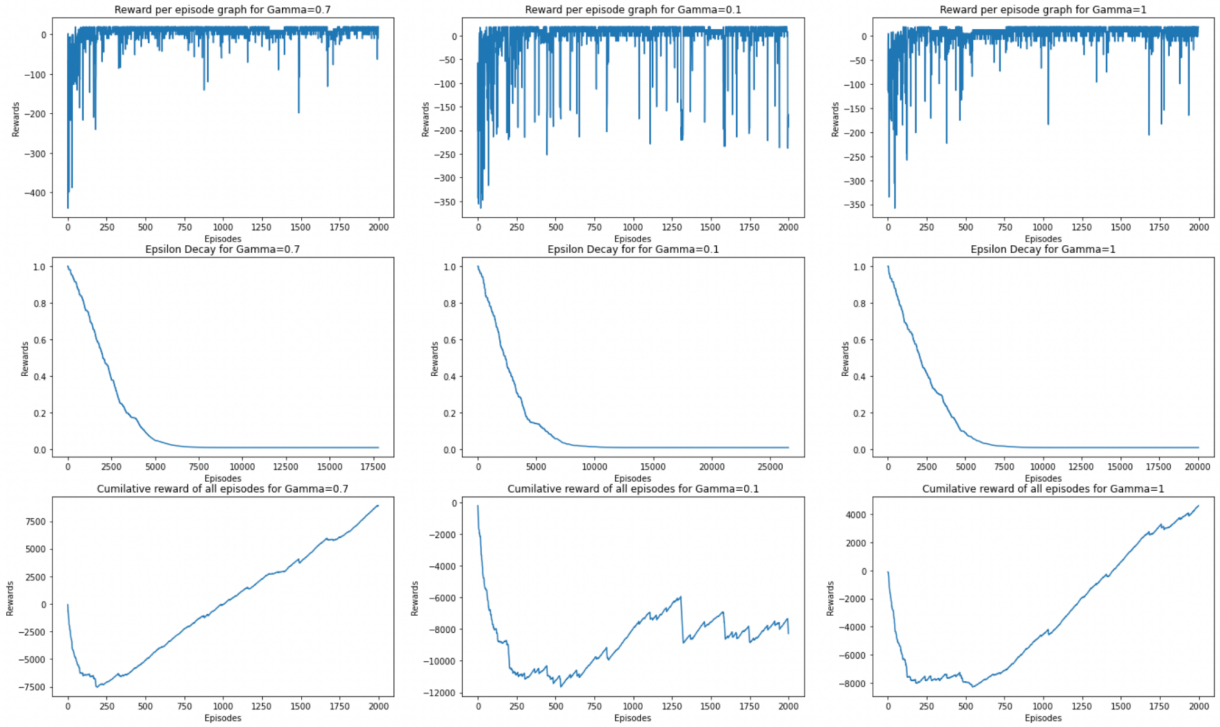
1. Changing Number of episodes values (num_of_episodes1=20, num_of_episodes2=200, num_of_episodes=2000)
2. The Column 1 of the plots show the reward per episode, Epsilon decay and Cumulative reward for no. of episodes=20. Similarly column 2 and column 3 are for no. of episodes values 200 and 2000 respectively.
3. From the graphs we can notice that the best value for no. of episodes is 200, For no. of episodes=20, the Q-learning algorithm didnt converge yet and Optimal policy is not yet found.
4. For no. of episodes=200 we have just converged and optimal policy has been found.
5. no. of episodes=2000 there is no point in running so many episodes, as the optimal policy is found around 200 episodes and it makes no difference training our agent for 2000 episodes.

C. Q-Learning Deterministic Environment (Varied Max time steps)



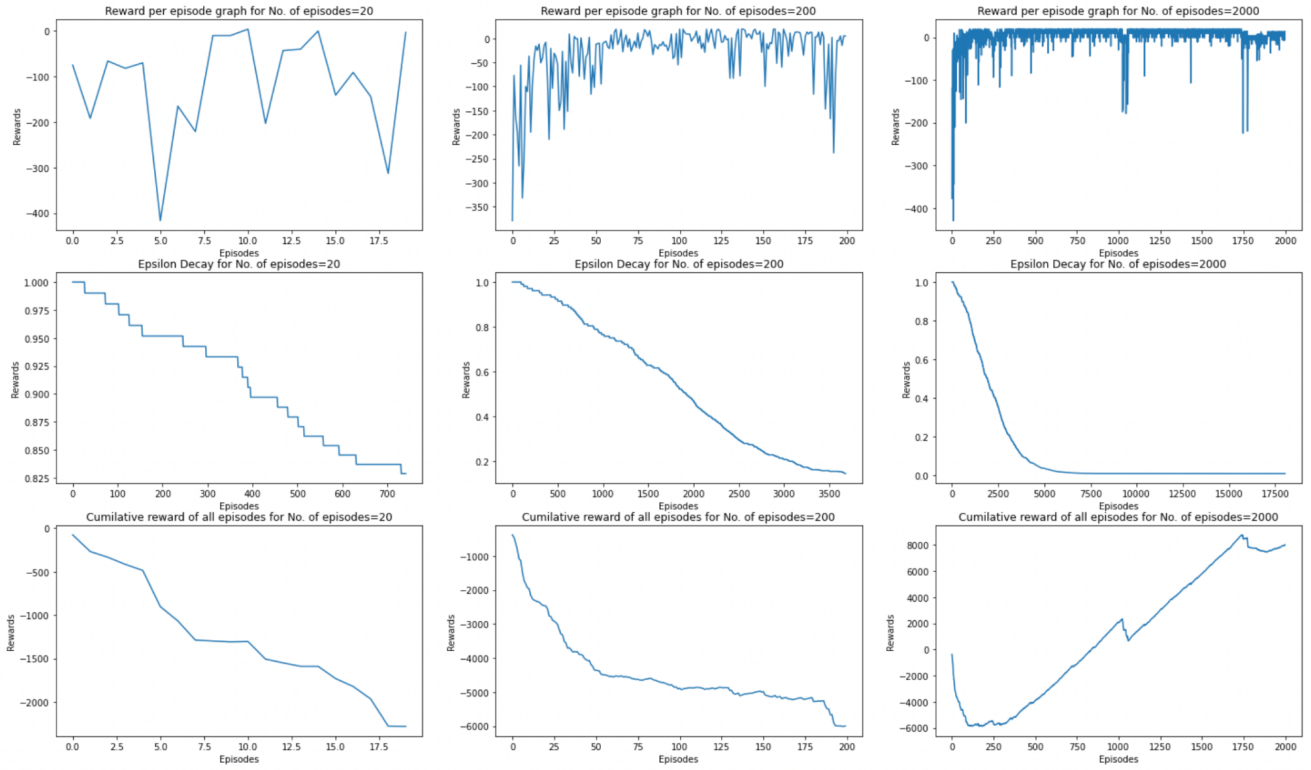
1. Changing Maximum time steps values (max_time_steps=10, num_of_episodes2=100, num_of_episodes=1000)
2. The Column 1 of the plots show the reward per episode, Epsilon decay and Cumulative reward for max time steps=10. Similarly column 2 and column 3 are for max time steps values 100 and 1000 respectively.
3. From the graphs we can notice that the best value for max time steps is 100, For max time steps=10, the Q-learning algorithm converges at around 75 episodes, but for max time steps=100, the optimal policy is found at around 50 max time steps.
4. Since our environment is relatively small, max time steps=1000 makes no sense.
5. Hence max time steps=100 is the best Parameter.

A. Q-Learning Stochastic Environment (Varied GAMMA)



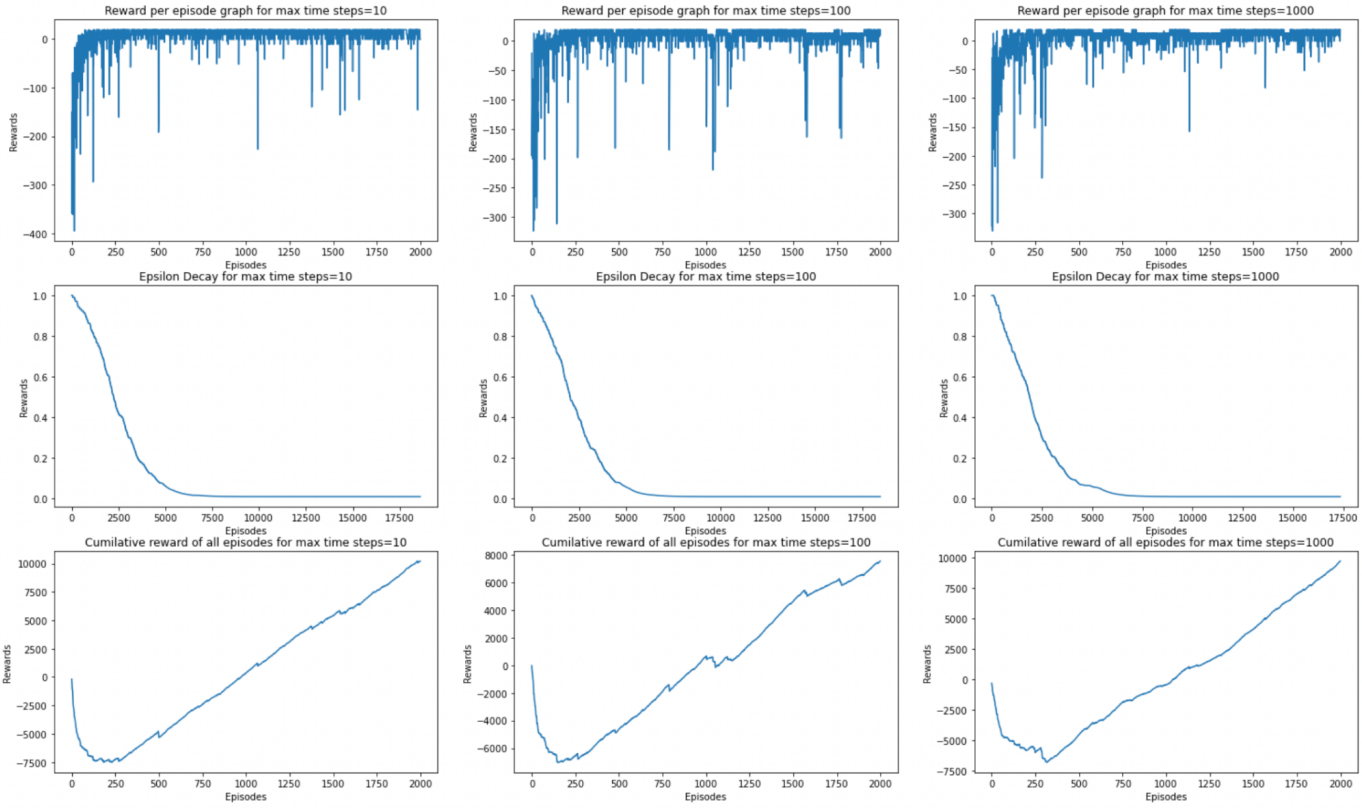
1. Changing Gamma values (gamma1=0.7, gamma2=0.1, gamma3=1)
2. For the given environment we have executed the Q-learning algorithm in a stochastic environment.
3. The Column 1 of the plots show the reward per episode, Epsilon decay and Cumulative reward for gamma=0.7. Similarly column 2 and column 3 are for gamma values 0.1 and 1 respectively.
4. From the graphs we can notice that the best value for Gamma is 0.7. Take a look at row 1 rewards per episode graph, it produces the best results for gamma=0.7

D. Q-Learning Stochastic Environment (Varied Number of Episodes)



1. Changing Number of episodes values (num_of_episodes1=20, num_of_episodes2=200, num_of_episodes=2000)
2. The Column 1 of the plots show the reward per episode, Epsilon decay and Cumulative reward for no. of episodes=20. Similarly column 2 and column 3 are for no. of episodes values 200 and 2000 respectively.
3. From the graphs we can notice that the best value for no. of episodes is 2000, For no. of episodes=20 and no. of episodes=200, the Q-learning algorithm didnt converge yet and Optimal policy is not yet found.
4. For no. of episodes around 220 we have just converged and optimal policy has been found.
5. no. of episodes=2000 there is no point in running so many episodes, as the optimal policy is found around 250 episodes and it makes no difference training our agent for 2000 episodes. Hence we need to adjust to 300 episodes possibly.

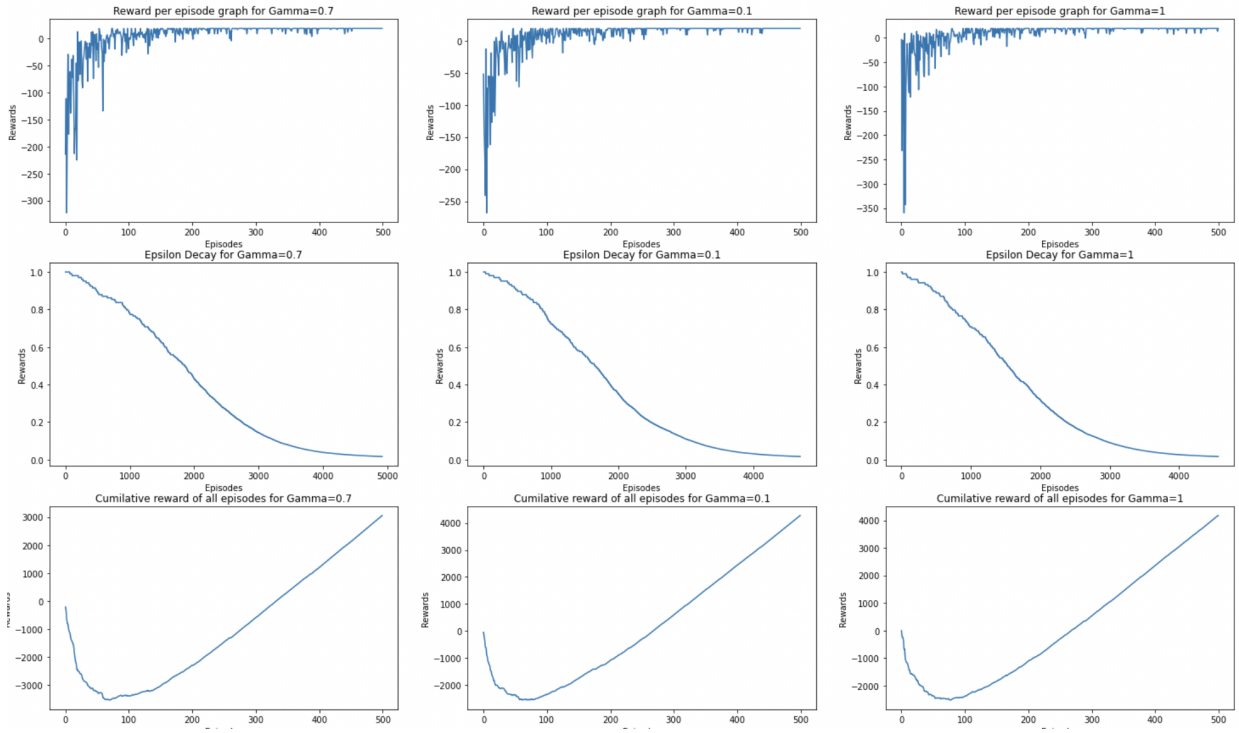
E. Q-Learning Stochastic Environment (Varied Max time steps)



6. Changing Maximum time steps values (max_time_steps=10, num_of_episodes2=100, num_of_episodes=1000)
7. The Column 1 of the plots show the reward per episode, Epsilon decay and Cumulative reward for max time steps=10. Similarly column 2 and column 3 are for max time steps values 100 and 1000 respectively.
8. From the graphs we can notice that the best value for max time steps is 100, For max time steps=10, the Q-learning algorithm converges at around 200 episodes, but for max time steps=100, the optimal policy is found a little early with sharp convergence.
9. Since our environment is relatively small, max time steps=1000 makes no sense. As it's a stochastic environment, it's also negatively impacting when max time steps are increased dramatically.
10. Hence max time steps=100 is the best Parameter.

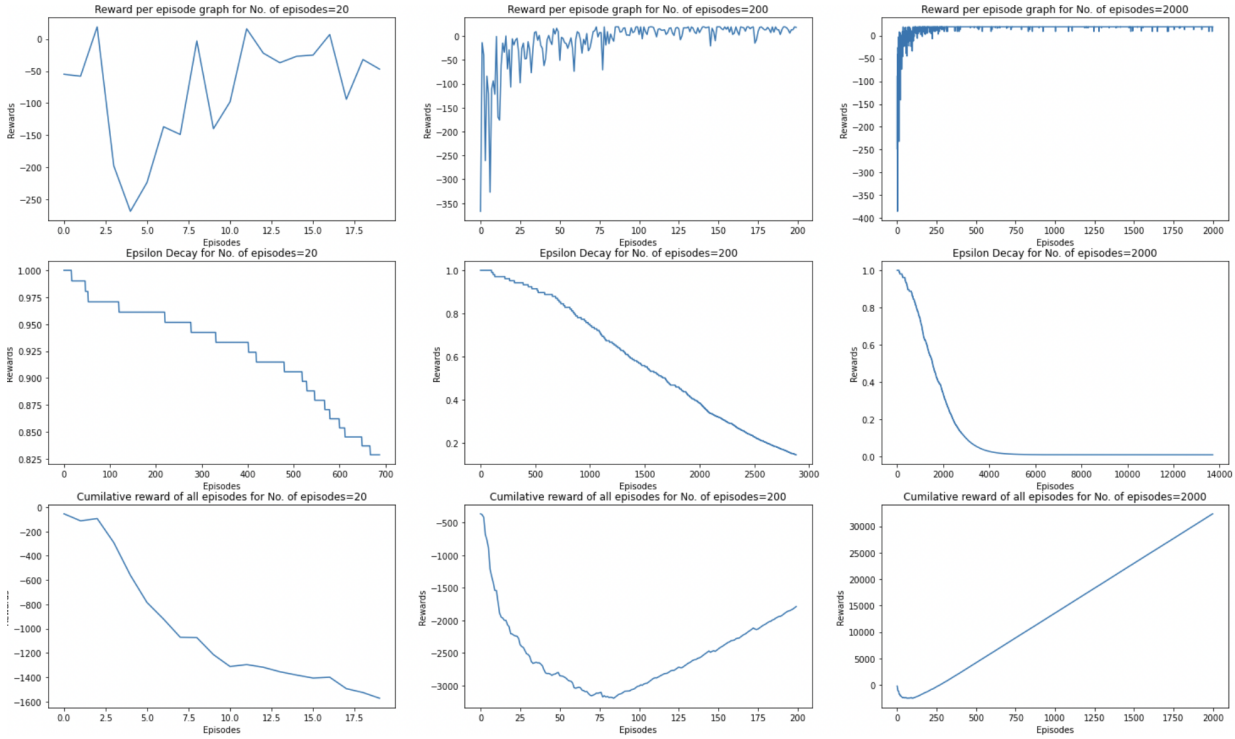
F. SARSA Deterministic Environment(Varied GAMMA)

1. Changing Gamma values (gamma1=0.7, gamma2=0.1, gamma3=1)



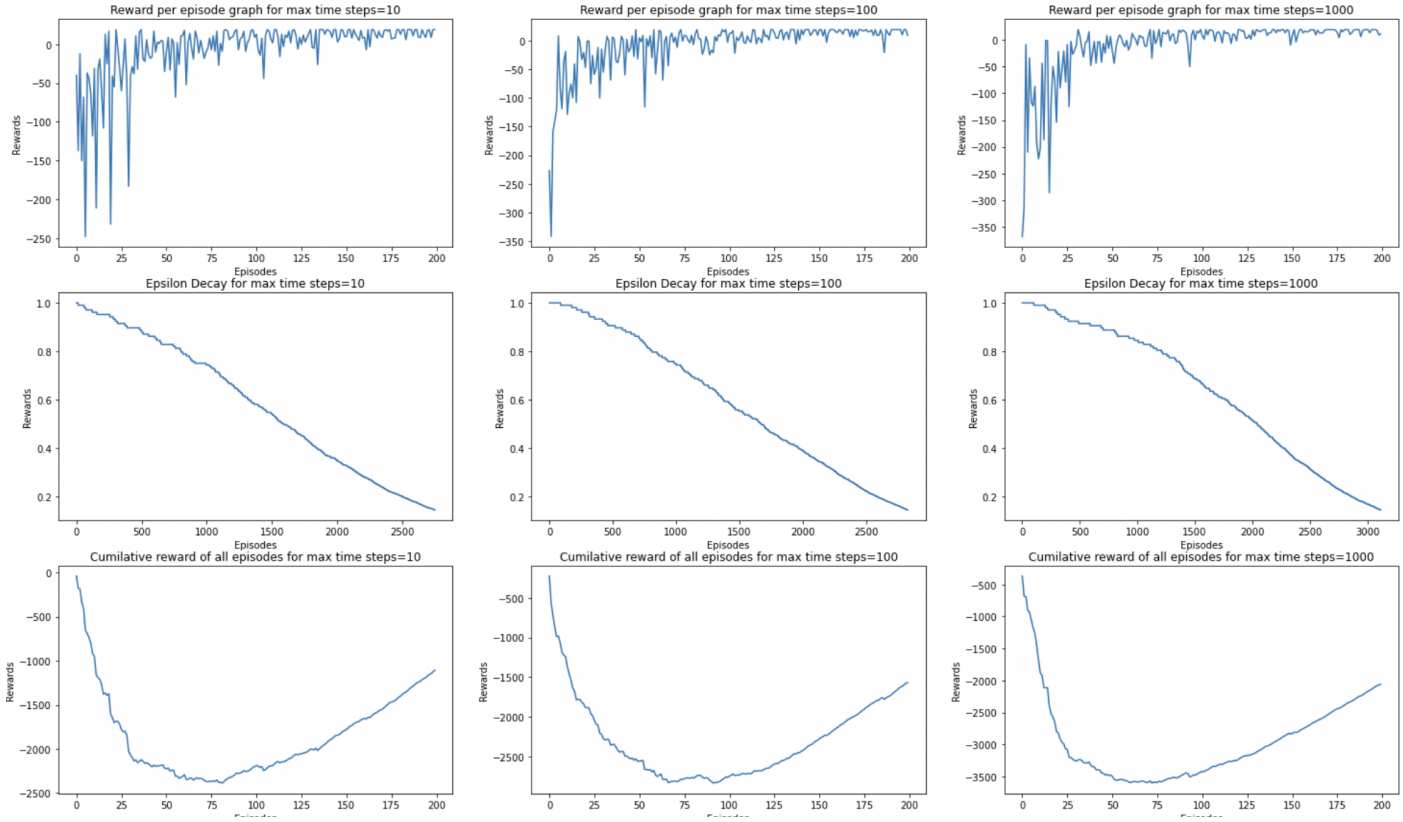
1. Changing Gamma values (gamma1=0.7, gamma2=0.1, gamma3=1)
2. For the given environment we have executed the SARSA algorithm in a deterministic environment.
3. The Column 1 of the plots show the reward per episode, Epsilon decay and Cumulative reward for gamma=0.7. Similarly column 2 and column 3 are for gamma values 0.1 and 1 respectively.
4. From the graphs we can notice that the best value for Gamma is 1. Take a look at row 1 rewards per episode graph, it produces the best results for gamma=1 and the optimal policy coverages smoother for gamma=1 than gamma=0.7.

G. SARSA Deterministic Environment (Varied Number of Episodes)



1. Changing Number of episodes values (num_of_episodes1=20, num_of_episodes2=200, num_of_episodes=2000)
2. The Column 1 of the plots show the reward per episode, Epsilon decay and Cumulative reward for no. of episodes=20. Similarly column 2 and column 3 are for no. of episodes values 200 and 2000 respectively.
3. From the graphs we can notice that the best value for no. of episodes is 200, For no. of episodes=20, the Q-learning algorithm did not converge yet and Optimal policy is not yet found.
4. For no. of episodes=200 we have just converged and optimal policy has been found.
5. no. of episodes=2000 there is no point in running so many episodes, as the optimal policy is found around 200 episodes and it makes no difference training our agent for 2000 episodes.

H. SARSA Deterministic Environment (Varied Max time steps)



I. Changing Maximum time steps values (max_time_steps=10, num_of_episodes2=100, num_of_episodes=1000)

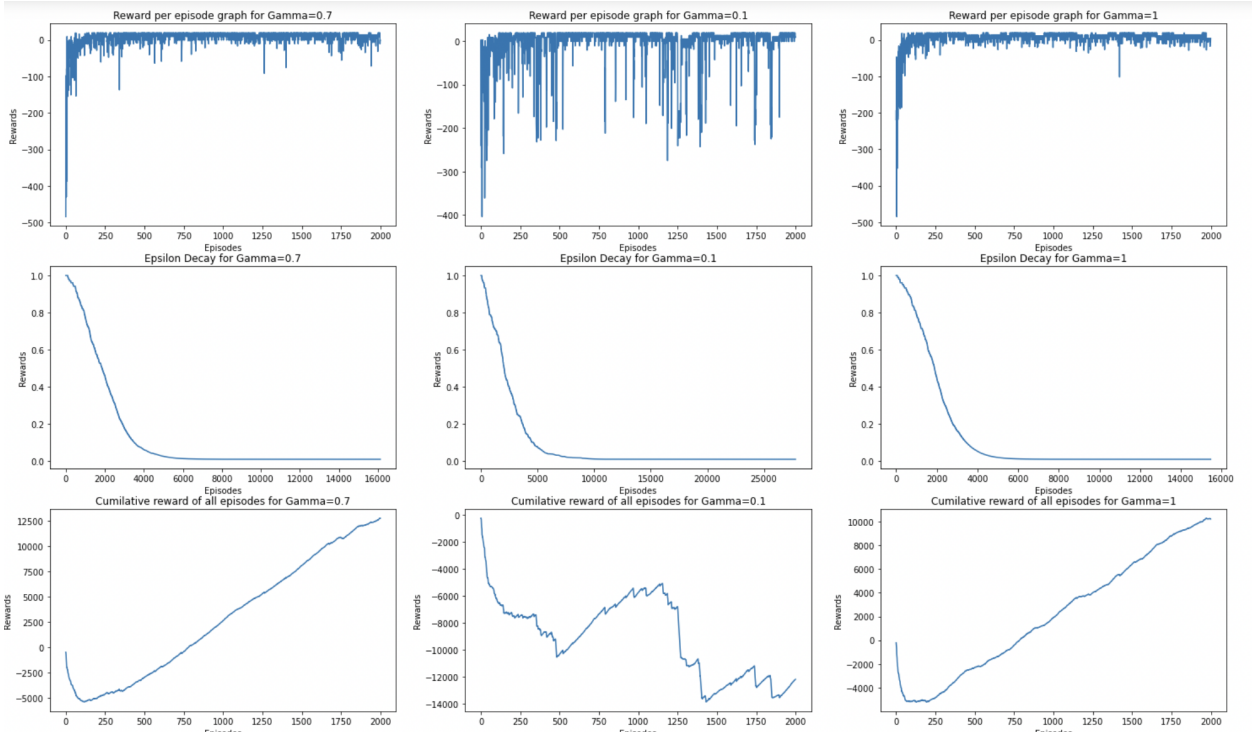
J. The Column 1 of the plots show the reward per episode, Epsilon decay and Cumulative reward for max time steps=10. Similarly column 2 and column 3 are for max time steps values 100 and 1000 respectively.

K. From the graphs we can notice that the best value for max time steps is 100, For max time steps=10, 100 and 1000. We can see that row one is optimal at max time steps =100.

L. Since our environment is relatively small, max time steps=1000 makes no sense.

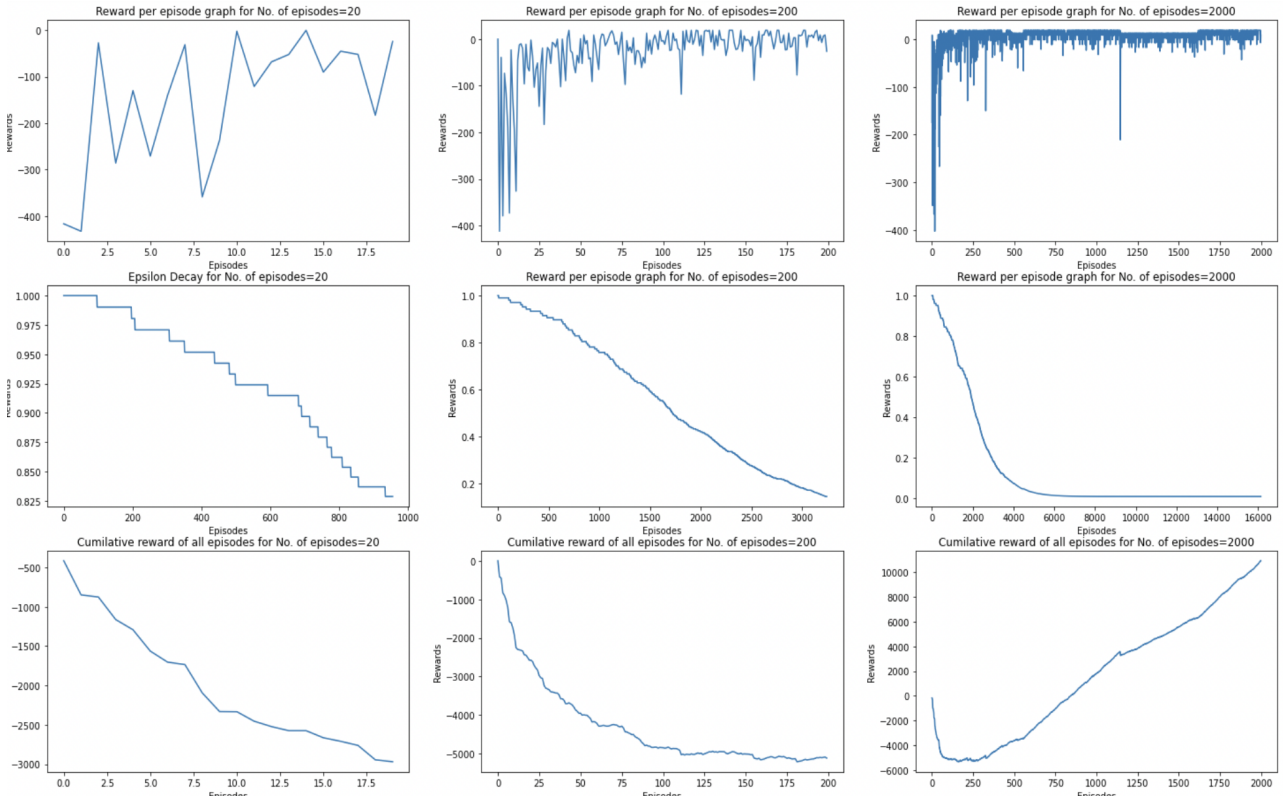
M. Hence max time steps=100 is the best Parameter.

N. SARSA Stochastic Environment(Varied GAMMA)



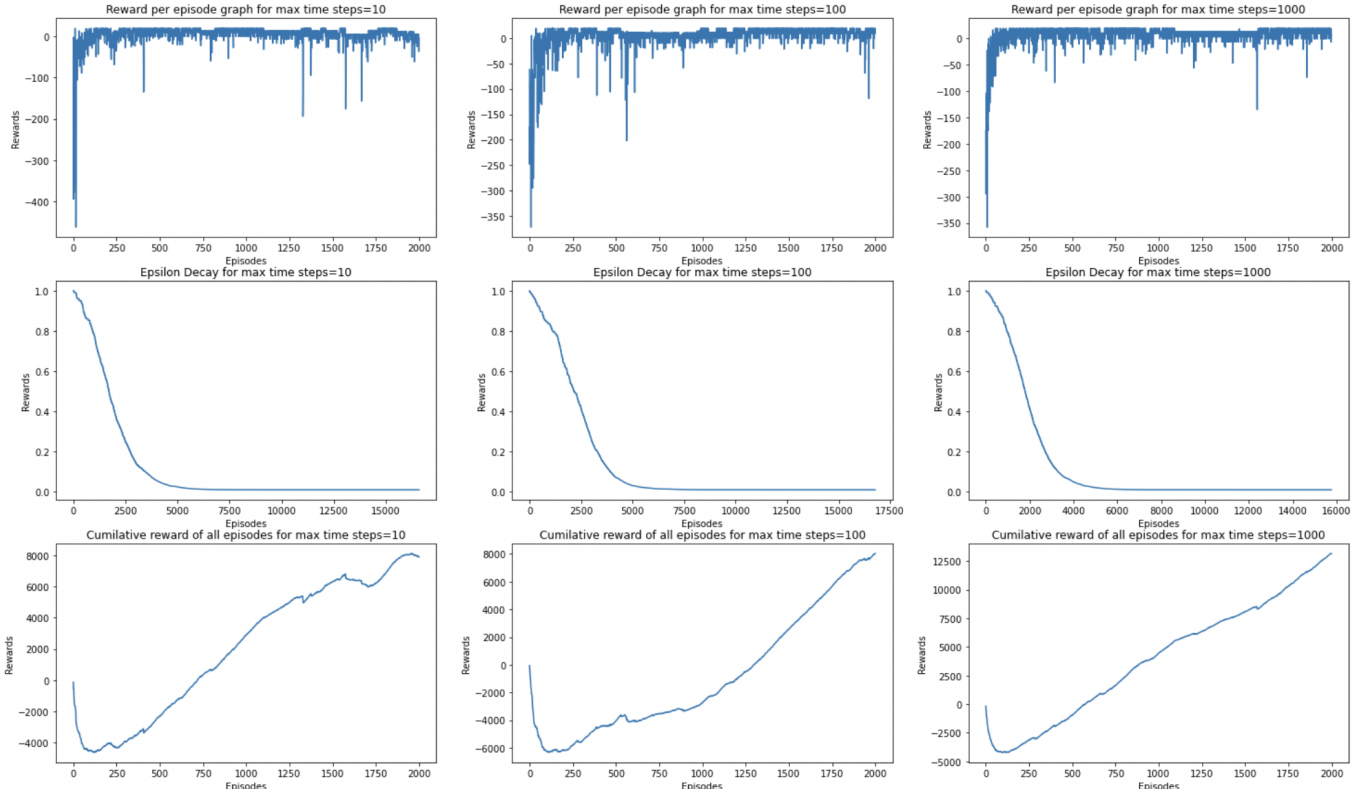
1. Changing Gamma values (gamma1=0.7, gamma2=0.1, gamma3=1)
2. For the given environment we have executed the SARSA algorithm in a stochastic environment.
3. The Column 1 of the plots show the reward per episode, Epsilon decay and Cumulative reward for gamma=0.7. Similarly column 2 and column 3 are for gamma values 0.1 and 1 respectively.
4. From the graphs we can notice that the best value for Gamma is 1. Take a look at row 1 rewards per episode graph and row 3 graphs, it produces the best results for gamma=1.

O. SARSA Stochastic Environment (Varied Number of Episodes)



6. Changing Number of episodes values (num_of_episodes1=20, num_of_episodes2=200, num_of_episodes=2000)
7. The Column 1 of the plots show the reward per episode, Epsilon decay and Cumulative reward for no. of episodes=20. Similarly column 2 and column 3 are for no. of episodes values 200 and 2000 respectively.
8. From the graphs we can notice that the best value for no. of episodes is 2000, For no. of episodes=20 and no. of episodes=200, the SARSA algorithm didnt converge yet and Optimal policy is not yet found.
9. For no. of episodes around 250 we have just converged and optimal policy has been found.
10. no. of episodes=2000 there is no point in running so many episodes, as the optimal policy is found around 250 episodes and it makes no difference training our agent for 2000 episodes. Hence we need to adjust to 300 episodes possibly.

P. SARSA Stochastic Environment (Varied Max time steps)



11. Changing Maximum time steps values (max_time_steps=10, num_of_episodes2=100, num_of_episodes=1000)
12. The Column 1 of the plots show the reward per episode, Epsilon decay and Cumulative reward for max time steps=10. Similarly column 2 and column 3 are for max time steps values 100 and 1000 respectively.
13. From the graphs we can notice that the best value for max time steps is 100, For max time steps=10, the SARSA algorithm converges at around 120 episodes, but for max time steps=100, the optimal policy is found a little early with sharp convergence.
14. Since our environment is relatively small, max time steps=1000 makes no sense. As it's a stochastic environment, it's also negatively impacting when max time steps are increased dramatically.
15. Hence max time steps=100 is the best Parameter.